

# ΚΕΦΑΛΑΙΟ 4

## *Εισαγωγή στις Αρχιτεκτονικές Σχεδίασης*

### *Περιεχόμενα*

- 4.1. Εισαγωγή στις Αρχιτεκτονικές Σχεδίασης
- 4.2. Σχεδιασμός Προσανατολισμένος στις διαδικασίες (Function Oriented)
- 4.3. Σχεδιασμός Προσανατολισμένος στα Αντικείμενα (Object Oriented)
- 4.4. Σχεδιασμός Διεπαφής Χρήστη

### *Διδακτικοί Στόχοι*

Στόχοι του κεφαλαίου αυτού είναι:

- Να περιγράφουν τα στάδια της φάσης σχεδιασμού συμπεριλαμβανομένης της αρχιτεκτονικής σχεδίασης και της σχεδίασης διεπαφών.
- Να χρησιμοποιούν τους διάφορους τύπους αρχιτεκτονικών στον σχεδιασμό πληροφοριακών συστημάτων.
- Να αξιοποιούν συγκεκριμένες αρχές και να υιοθετούν καλές πρακτικές για να σχεδιάζουν αποτελεσματικά τη γραφική διεπαφή χρήστη.
- Να εκτιμούν την αναγκαιότητα του σχεδιασμού αρχιτεκτονικής και λογισμικού.

### 1.1. Εισαγωγή στις Αρχιτεκτονικές Σχεδίασης

Όσο το μέγεθος και η πολυπλοκότητα των πληροφοριακών συστημάτων μεγαλώνει ραγδαία, το πρόβλημα της σχεδίασης του λογισμικού και δομών δεδομένων δεν αποτελούν πλέον κύρια προβλήματα σχεδίασης. Όταν τα συστήματα αποτελούνται από πολλά δομικά στοιχεία (components), η οργάνωση του συνολικού συστήματος αποτελεί την νέα σχεδιαστική πρόκληση.

Για την σχεδίαση τέτοιων συστημάτων επιστρατεύονται πολλοί και διάφοροι τρόποι, τεχνοτροπίες και μέθοδοι, οι οποίες περιλαμβάνουν άτυπα διαγράμματα και περιγραφικούς όρους, γλώσσες παραμέτρων σύνδεσης των τμημάτων, πρότυπα και τυπικά μοντέλα μηχανισμών ολοκλήρωσης των διαφορετικών τμημάτων.

Όταν αναφερόμαστε στα δομικά προβλήματα σχεδίασης εννοούμε την δυναμική οργάνωση και την ολική δομή επικοινωνίας των μερών του συστήματος, πρωτόκολλα επικοινωνίας, θέματα συγχρονισμού των μερών, ανάθεση λειτουργικότητας ξεχωριστά σε κάθε τμήμα του συστήματος, κατανομή φυσικών πόρων, σύνθεση των τμημάτων και τέλος αξιολόγηση των διαφόρων εναλλακτικών αρχιτεκτονικών σχεδίασης.

Όλα αυτά συνιστούν το επίπεδο της αρχιτεκτονικής σχεδίασης του συστήματος. Είναι προφανές ότι η σωστή κατασκευή ενός συστήματος προϋποθέτει την σωστή αρχιτεκτονική της σχεδίασης του, η οποία θα αποτελέσει την βάση. Επίσης :

1. Πρέπει να κατηγοριοποιηθούν τα υπάρχοντα συστήματα των οποίων τα κοινά χαρακτηριστικά θα μπορούν να χρησιμοποιηθούν ως πρότυπα για την σχεδίαση νέων συστημάτων.
2. Η σωστή επιλογή αρχιτεκτονικής συστήματος πρέπει να γίνει από την αρχή γιατί μπορεί να οδηγήσει σε καταστροφικά αποτελέσματα.
3. Λεπτομερής κατανόηση των αρχιτεκτονικών σχεδίασης των συστημάτων επιτρέπει στον υπεύθυνο σχεδιασμού να κάνει τις κατάλληλες επιλογές μεταξύ εναλλακτικών επιλογών σχεδίασης.
4. Η παρουσίαση της αρχιτεκτονικής ενός συστήματος είναι ουσιαστική για την ανάλυση και περιγραφή του υψηλότερου επιπέδου ενός σύνθετου συστήματος.

**Η Αρχιτεκτονική Σχεδίαση (ΑΣ)** περιγράφει τον τρόπο λειτουργίας και χρήσης τμημάτων του συστήματος, μας βοηθάει να καταλάβουμε πως θα έπρεπε να οργανωθεί και να σχεδιασθεί η συνολική δομή του πληροφοριακού συστήματος. Είναι το πρώτο πράγμα που πρέπει να κάνουμε κατά την διαδικασία της σχεδίασης και αποτελεί την γέφυρα μεταξύ της σχεδίασης και της διαδικασίας κατασκευής του λογισμικού. Το αποτέλεσμα της αρχιτεκτονικής σχεδίασης είναι ένα αρχιτεκτονικό

μοντέλο το οποίο περιγράφει πως το όλο σύστημα είναι οργανωμένο ως ένα σύνολο από κομμάτια που επικοινωνούν μεταξύ τους.

Κατά την Ευέλικτη Διαδικασία (Agile Process) είναι γενικά αποδεκτό ότι το πρώτο στάδιο, όπως είναι η σχεδίαση, θα έπρεπε να ασχολείται με το να δώσει μια συνολική εικόνα του συστήματος. Αντίθετα η μέθοδος της Σταδιακής Ανάπτυξης, με την οποία ξεκινάμε την σχεδίαση από ένα αρχικό τμήμα και στην πορεία συνδέουμε καινούργια τμήματα, αποδείχθηκε στην πορεία ένας λάθος τρόπος προσέγγισης. Η τρίτη επιλογή που έχουμε είναι να γίνει μια αρχική σχεδίαση και στην πορεία να κάνουμε διορθωτικές επεμβάσεις. Ο τρόπος αυτός αν και είναι εύκολος, έχει αρκετά μεγάλο κόστος.

Στην πράξη, υπάρχει σημαντική επικάλυψη μεταξύ της διαδικασίας εύρεσης των απαιτήσεων του συστήματος και της αρχιτεκτονικής σχεδίασης. Κανονικά η **καταγραφή προδιαγραφών** του συστήματος **δεν πρέπει** να περιέχει στοιχεία σχεδίασης. Αυτό επιτρέπεται μόνον στον σχεδιασμό μικρών συστημάτων. Η αρχιτεκτονική σχεδίαση έχει σαν αποστολή να βοηθήσει στην οργάνωση και δόμηση των προδιαγραφών (System Specifications). Συνεπώς ο αναλυτής του συστήματος, θα μπορούσε να προτείνει ως μέρος του ορισμού των απαιτήσεων του συστήματος, μια ΑΣ, όπου θα υπάρχουν οι συσχετισμοί μεταξύ των λειτουργιών του συστήματος ή χαρακτηριστικών μεταξύ των μεγάλων ή μικρότερων τμημάτων.

Ο σχεδιασμός της αρχιτεκτονικής σχεδίασης ανάλογα με το μέγεθος της, μπορεί να είναι δύο κατηγοριών.

- **Αρχιτεκτονική Σχεδίαση Μικρής Κλίμακας.** Αυτή ασχολείται με την ΑΣ μικρών τμημάτων του συστήματος. Το επίπεδο αυτό αφορά τα μικρότερα τμήματα από τα οποία αποτελείται ένα βασικό τμήμα του συστήματος. Η βάση δίνεται στην αρχιτεκτονική προγραμμάτων και τοπικών λειτουργιών.
- **Αρχιτεκτονική Σχεδίαση Μεγάλης Κλίμακας.** Αυτή ασχολείται με πολύ-σύνθετα συστήματα επιχειρήσεων και οργανισμών τα οποία περιέχουν και άλλα συστήματα, προγράμματα και κομμάτια άλλων προγραμμάτων. Αυτά τα διάφορα συστήματα είναι καταναμεμημένα σε διαφορετικούς υπολογιστές και ενδεχομένως να τα υποστηρίζουν διαφορετικές και ανεξάρτητες μεταξύ τους εταιρείες ή οργανισμοί.

#### 4.1.1. Τεχνοτροπίες Σχεδίασης

Οι τεχνοτροπίες αυτές είναι προσανατολισμένες στις εξής δυο βασικές κατηγορίες :

1. **Σχεδίαση βασισμένη στις Διαδικασίες (Function Oriented)**, όπου γίνεται διάκριση δεδομένων από τις διαδικασίες (Δομημένη Σχεδίαση).
2. **Σχεδίαση βασισμένη στα αντικείμενα (Object Oriented)**, όπου οι οντότητες και οι διαδικασίες ενσωματώνονται στο ίδιο κέλυφος. Η συλλογή των

οντοτήτων που περικλείουν τα δεδομένα και τις διαδικασίες παρέχουν το σύνολο των υπηρεσιών.

Ο σχεδιαστής και στις δυο περιπτώσεις ασχολείται και με τα δεδομένα και με τις διαδικασίες. Αναλυτική παρουσίαση των δύο αυτών τεχνοτροπιών παρουσιάζονται στα παρακάτω κεφάλαια.

Οι παραπάνω τεχνοτροπίες που εξετάζονται αναλυτικά στη συνέχεια, συχνά αναφέρονται και σαν ιεραρχικά διαγράμματα, γιατί έχουν τη μορφή δέντρου. Για τον ίδιο λόγο και οι τεχνικές αυτές ονομάζονται και δομημένες διαγραμματικές τεχνικές.

#### 4.1.2. Είδη Σχεδίασης

Σχετικά με τα είδη σχεδίασης, έχουμε τα εξής:

- **Αρχιτεκτονική Σχεδίαση**

Στη σχεδίαση αυτή γίνεται καθορισμός μονάδων του λογισμικού και διάταξής τους στο σύστημα

- **Σχεδίαση Διαπροσωπειών (Interfaces)**

Εδώ κάνουμε τον καθορισμό της επικοινωνίας των μονάδων του συστήματος με :

- Άλλα συστήματα
- Συσκευές
- Χρήστες

- **Λεπτομερής Σχεδίαση Μονάδων**

Στο επίπεδο αυτό γίνεται καθορισμός της δομής κάθε μονάδας του συστήματος

- **Σχεδίαση Δεδομένων**

Λεπτομερή περιγραφή και σχεδίαση των δεδομένων

### 1.2. Αρχιτεκτονική Σχεδίασης προσανατολισμένη στις Διαδικασίες (Function Oriented)

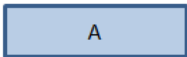


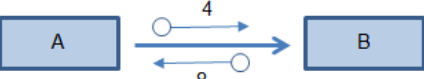
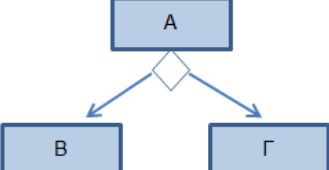
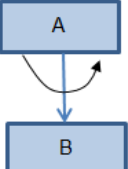
Ας δούμε την διαδικασία κατασκευής του αρχιτεκτονικού σχεδίου ενός συστήματος μιας σχολικής βιβλιοθήκης, βασιζόμενοι στο διάγραμμα ροής δεδομένων του προηγούμενου κεφαλαίου.

#### 1.2.1 Διάγραμμα Δομής Συστήματος

Η διαγραμματική απεικόνιση είναι ένα πολύ σημαντικό βήμα για την αποτύπωση του σχεδιαστικού μοντέλου του συστήματός μας.

Το διάγραμμα δομής βασίζεται στο διάγραμμα ροής δεδομένων (ΔΡΔ), από το οποίο μέσα από κάποια διαδοχικά βήματα μπορούμε να οδηγηθούμε στο διάγραμμα δομής (ΔΔ). Στα διαγράμματα αυτά οι μονάδες λογισμικού παριστάνονται με ένα παραλληλόγραμμο (α), όπως φαίνεται στο Σχήμα 4.1.

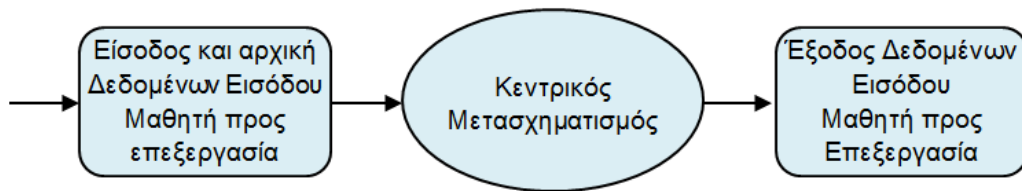
Η κάθε μονάδα του συστήματος μπορεί να καλέσει κάποια άλλη, και η αποτύπωση της κλήσης αυτής είναι το βέλος (β). Η αποτύπωση της αποστολής ή παραλαβής δεδομένων από τις μονάδες γίνεται με το βέλος με έναν κύκλο στο άκρο της αρχής του (γ) όταν γίνεται με μια κατεύθυνση ή με τα διπλά βέλη όταν υπάρχει αμφίδρομη αποστολή δεδομένων.

(α) Μονάδα Λογισμικού (Λειτουργία Α)	
(β) Σύνδεση Μονάδων (Από Α στην Β)	
(γ) Πέρασμα Παραμέτρων (Αριθμός 4) από Λειτουργία Α στην Β	
(δ) Πέρασμα Παραμέτρου (Αριθμός 4) από Λειτουργία Α στην Β και πέρασμα παραμέτρου (Αριθμός 8) από Λειτουργία Β στην Α	
(ε) Μονάδα η οποία αποτελεί σημείο ελέγχου ροής σε άλλες μονάδες	
(ζ) Η Μονάδα επαναληπτικής δομής, συμβολίζεται με το ημι- κυκλικό βέλος. Υποδηλώνει ότι η μονάδα αυτή επαναλαμβάνεται πολλές φορές.	

**Σχήμα 4.1:** Συμβολισμοί Διαγράμματος Δομής

Στα διαγράμματα αυτά υπάρχουν δύο βασικοί τύποι χαρακτηριστικών περιοχών. Αυτά είναι οι **κεντρικοί μετασχηματισμοί** και τα **κέντρα δοσοληψιών**.

**Κεντρικός Μετασχηματισμός** είναι ένα σύνολο επεξεργασιών που εκτελεί την επεξεργασία δεδομένων εισόδου και παράγει δεδομένα εξόδου. Τα δεδομένα εισόδου ετοιμάζονται από ένα σύνολο επεξεργασιών που βρίσκονται πριν από τον κεντρικό μετασχηματισμό. Τα δεδομένα εξόδου ετοιμάζονται από επεξεργασίες που ακολουθούν τον κεντρικό μετασχηματισμό Σχήμα 4.2.



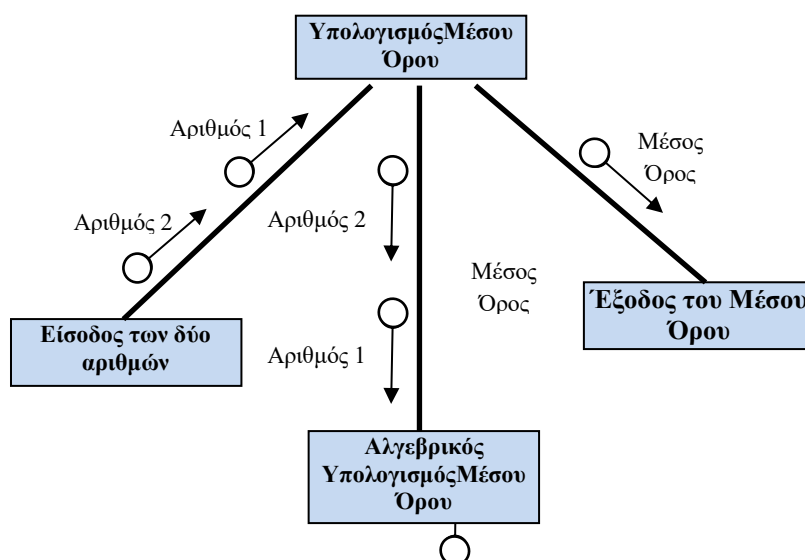
**Σχήμα 4.2:** Κεντρικός Μετασχηματισμός Σχολικής Βιβλιοθήκης

Να αναφέρουμε ότι οι μετασχηματισμοί (λειτουργίες) του παραπάνω σχήματος, ενδέχεται να είναι θύλακες άλλων εσωτερικών μετασχηματισμών. Συνεπώς, η Είσοδος και αρχική επεξεργασία Δεδομένων Εισόδου Μαθητή, όπως και η Επεξεργασία και η έξοδος Δεδομένων Εισόδου Μαθητή, μπορούν να διαιρεθούν σε άλλα σύνθετα τμήματα του διαγράμματος ροής δεδομένων και όχι σε απλούς, αυτόνομους μετασχηματισμούς.

Ο εντοπισμός και η πιθανή ανάλυση των μετασχηματισμών, δεν είναι μια αυτοματοποιημένη διαδικασία βημάτων την οποία θα μπορούσαμε να ακολουθήσουμε. Για να γίνει αυτό πρέπει να επιστρατεύσουμε την αναλυτική μας ικανότητα, την εμπειρία και τον αυτοσχεδιασμό μας. Το βασικό για την καλή δημιουργία των διαγραμμάτων δομής είναι η σύλληψη και η λεπτομέρεια του διαγράμματος ροής στο οποίο στηρίζεται.

### 1.2.2 Πρώτο Παράδειγμα

Ας θεωρήσουμε ένα απλό παράδειγμα για να καταλάβουμε την δομή ενός διαγράμματος δομής λειτουργιών ενός συστήματος. Έστω ότι θέλουμε να σχεδιάσουμε ένα σύστημα το οποίο να δέχεται δύο αριθμούς και να εξάγει τον μέσο όρο τους.



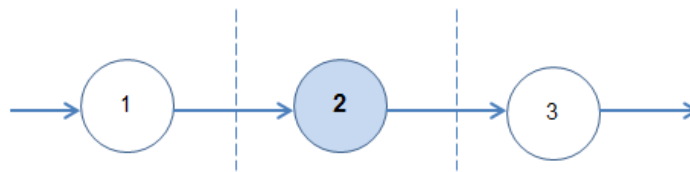
**Σχήμα 4.3:** Διάγραμμα Δομής Υπολογισμού Μέσου Όρου

Αν και το παραπάνω παράδειγμα αφορά την δομή μιας προγραμματιστικής μονάδας, και όχι ενός πληροφοριακού συστήματος, η θεώρησή του είναι βασική για την κατανόηση των συμβόλων και του τρόπου λειτουργίας του ΔΔ. Βλέπουμε στο σχήμα 4.3, ότι έχουμε έναν κεντρικό μετασχηματισμό (Υπολογισμός Μέσου Όρου) και δυο υπό-μονάδες για την είσοδο των δύο αριθμών, και την έξοδο του δεδομένου εξόδου. Είναι εύκολο να καταλάβουμε και τον συμβολισμό της μετακίνησης των δεδομένων.

### 1.2.3 Δεύτερο Παράδειγμα

Στο παράδειγμα αυτό θα δούμε την σχεδίαση ενός διαγράμματος δομής δεδομένων το οποίο αφορά την σχολική βιβλιοθήκη προηγούμενου κεφαλαίου.

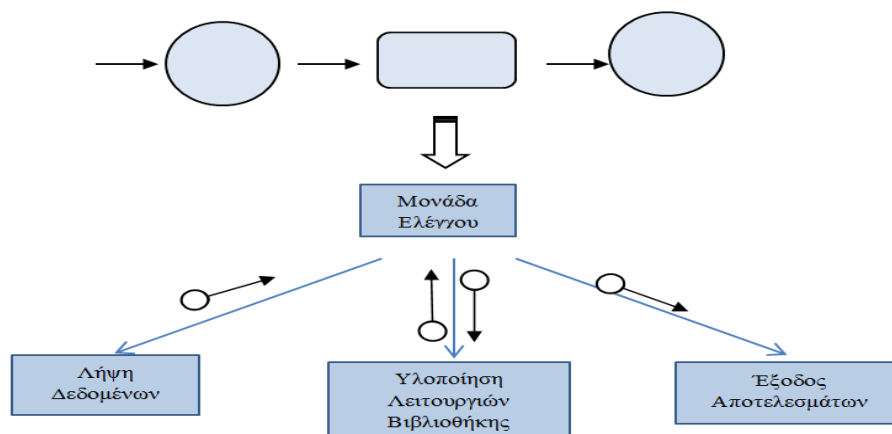
Ας αρχίσουμε με μια εναλλακτική απεικόνιση του κεντρικού μετασχηματισμού του σχήματος 4.4.



**Σχήμα 4.4:** Διάγραμμα Κεντρικού Μετασχηματισμού

Το διάγραμμα του σχήματος 4.4. αποτυπώνει την εντελώς βασική απεικόνιση του κεντρικού μετασχηματισμού. Η μονάδα ελέγχου εκτέλεσης (2), καλεί την μονάδα που απαιτείται για την είσοδο των δεδομένων (1) και την μονάδα (3) για την έξοδο των δεδομένων.

Στο σχήμα 4.5. βλέπουμε την μετατροπή του βασικού κεντρικού μετασχηματισμού σε διάγραμμα δομής, όπου φαίνονται οι απλοί μετασχηματισμοί που αντιστοιχούν στις μονάδες του συστήματος.



**Σχήμα 4.5:** Αντιστοίχιση Κεντρικού Μετασχηματισμού σε διάγραμμα δομής

**Κέντρο Δοσοληψιών** (transaction center). Σε ένα διάγραμμα ροής δεδομένων μια επεξεργασία χαρακτηρίζεται ως κέντρο δοσοληψιών όταν δέχεται δεδομένα από επεξεργασίες που τα προετοιμάζουν και βρίσκονται πριν από αυτό και διανέμει τη ροή σε διάφορες επεξεργασίες ανάλογα με την λογική του συστήματος.

Η διαφορά του με τον κεντρικό μετασχηματισμό είναι στο ότι είναι αδιαίρετη μονάδα όσον αφορά την λειτουργία του. Δηλαδή, ενώ ένας κεντρικός μετασχηματισμός μπορεί να αναλυθεί σε περισσότερους του ενός μετασχηματισμούς του διαγράμματος ροής δεδομένων, ένα κέντρο δοσοληψιών περιλαμβάνει μόνο μια λειτουργία η οποία δεν αναλύεται σε άλλες απλούστερες. Ένα κέντρο δοσοληψιών φαίνεται στο Σχήμα 4.6.



**Σχήμα 4.6:** Ανάλυση Κεντρικού Μετασχηματισμού

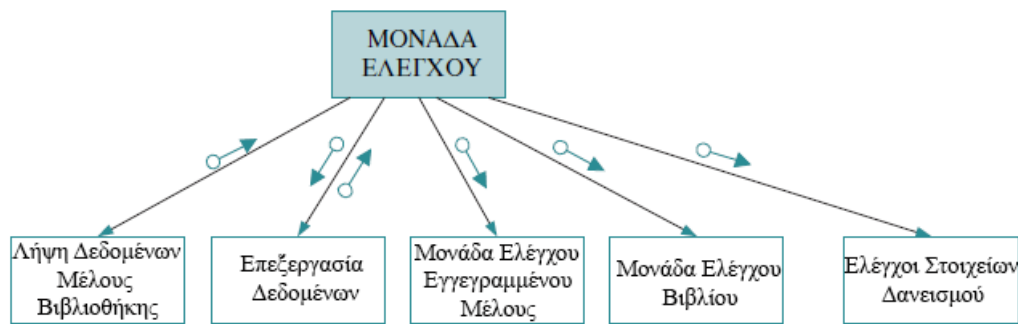
Χαρακτηριστική περίπτωση κέντρου δοσοληψιών είναι ένας μετασχηματισμός ελέγχου ροής προγράμματος, όπου, ανάλογα με την επιλογή του χρήστη, η ροή μεταφέρεται σε διαφορετικούς μετασχηματισμούς, όπως σε ένα μενού (βλέπε Σχήμα 4.7).

Στο σχήμα 4.7, η μονάδα ελέγχου του κέντρου δοσοληψιών καλεί δύο μονάδες για τη λήψη των δεδομένων εισόδου, και επεξεργασία αντίστοιχα, και μεταφέρει τον έλεγχο σε τόσες μονάδες, όσες είναι οι περιπτώσεις των δεδομένων εξόδου.

#### 1.2.4 Βήματα κατασκευής διαγραμμάτων δομής

Για να μετατρέψουμε ένα διάγραμμα ροής σε ένα διάγραμμα δομής κάνουμε διαδοχικό διαμελισμό των λειτουργιών, με διάσχιση κατά πλάτος (BreathFirst), μέχρι να προσδιοριστούν και να απεικονιστούν όλοι οι μετασχηματισμοί που περιέχονται στο διάγραμμα ροής του συστήματος.

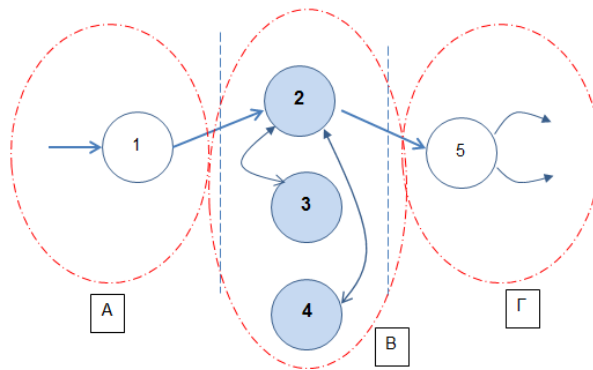




*Σχήμα 4.7: Κέντρο Δοσοληψίας σε διάγραμμα δομής*

Τα βήματα αυτά είναι :

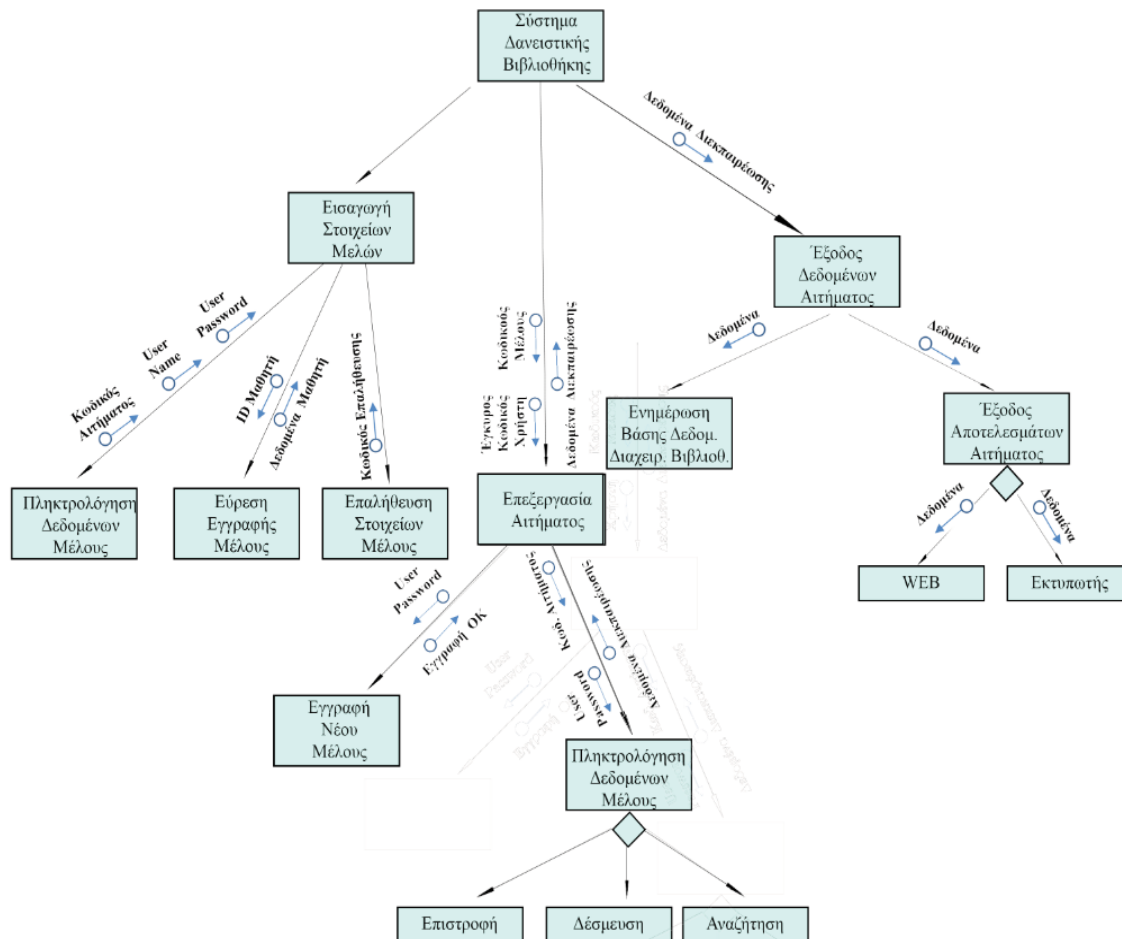
- **Εντοπίζουμε τον κεντρικό μετασχηματισμό.** Για κάθε τμήμα του ΔΡΔ, εντοπίζουμε τον κεντρικό μετασχηματισμό, και τους μετασχηματισμούς εισόδου και εξόδου.
- **Μετατρέπουμε τον κεντρικό μετασχηματισμό σε διάγραμμα δομής.** Δημιουργούμε το πρώτο επίπεδο του διαγράμματος δομής (ΔΔ) που αντιστοιχεί στον κεντρικό μετασχηματισμό.
- **Παραγοντοποίηση (factoring).** Κατασκευάζουμε τα διαγράμματα δομής για τους μετασχηματισμούς εισόδου και εξόδου, αριστερά, και δεξιά του κεντρικού. Κάθε ένας από αυτούς περιέχει μια μονάδα ελέγχου η οποία παραλαμβάνει τα δεδομένα εισόδου και μια άλλη μονάδα ελέγχου η οποία τα εξάγει στους αποδέκτες των δεδομένων, όπως είναι ο χρήστης, τα εξωτερικά συστήματα, οι συσκευές εξόδου (εκτυπωτής) και τα αρχεία. Η παραγοντοποίηση είναι μια αυτο-επαναλαμβανόμενη λειτουργία (recursive) για κεντρικούς μετασχηματισμούς που θα αναγνωριστούν στην πορεία με αποτέλεσμα να γίνει η ίδια διαδικασία ανάλυσης και για αυτούς. Η διαδικασία επαναλαμβάνεται μέχρι την πλήρη ανάλυση του συστήματος.
- **Συνένωση.** Για περιπτώσεις όπου τα δεδομένα εισόδου ή εξόδου δεν λαμβάνονται από τον χρήστη ή εξωτερικά συστήματα, τότε οι μονάδες αυτές αντικαθίστανται από μονάδες έλεγχου οι οποίες παρέχουν τα δεδομένα αυτά. Αν ο καλύτερος σχεδιασμός του διαγράμματος δομής απαιτεί διορθώσεις στο επίπεδο της ανάλυσης, το οποίο είναι το διάγραμμα ροής, τότε αυτό είναι χρήσιμο να γίνει τώρα και όχι αργότερα.



**Σχήμα 4.8:** Κεντρικός Μετασχηματισμός για Σχολική Βιβλιοθήκη

Τα τμήματα 2, 3 και 4 αποτελούν μέρη του κεντρικού μετασχηματισμού. Στη συνέχεια βασιζόμενοι στη σχήμα 4.8 δημιουργούμε την απεικόνιση του διαγράμματος δομής. Στο τμήμα Α, γίνεται εισαγωγή των στοιχείων του υποψήφιου χρήστη ή του υπάρχοντος μέλους της βιβλιοθήκης. Στη συνέχεια πληκτρολογούνται τα στοιχεία του και μεταφέρονται από την μονάδα ελέγχου Εισαγωγή Στοιχείων Μελών για εύρεση αν είναι ήδη μέλος της βιβλιοθήκης, και επιστρέφονται τα δεδομένα του (Όνομα, Βιβλία που δανείστηκε κλπ). Στη συνέχεια γίνεται επαλήθευση των στοιχείων του με χρήση της Αποθήκης Δεδομένων (ΒΔ).

Στο επόμενο βήμα, γίνεται επεξεργασία του αιτήματος του χρήστη/μέλους, η οποία είτε οδηγεί σε έγγραφη νέου μέλους, είτε οδηγεί σε διεκπεραίωση αιτήματος ήδη υπάρχοντος μέλους (Επιστροφή βιβλίου, Δέσμευση βιβλίου, Αναζήτηση Τίτλου κλπ). Όταν τελειώσει, τα δεδομένα – αποτελέσματα, προωθούνται από την κεντρική μονάδα ελέγχου στη μονάδα εξόδου. Γίνεται ενημέρωση της αποθήκης της βιβλιοθήκης και ενημερώνεται η εγγραφή του νέου ή παλιού μέλους με το αίτημα που είχε, και τα αποτελέσματά του. Επίσης η μονάδα εξόδου προωθεί τα αποτελέσματα του αιτήματος (Transaction), στο web μέσα από κάποια διαδικτυακή διεπαφή, είτε τα προωθεί στον εκτυπωτή αν κριθεί από τον διαχειριστή εμπρόσθιου άκρου (Front End Administrator).



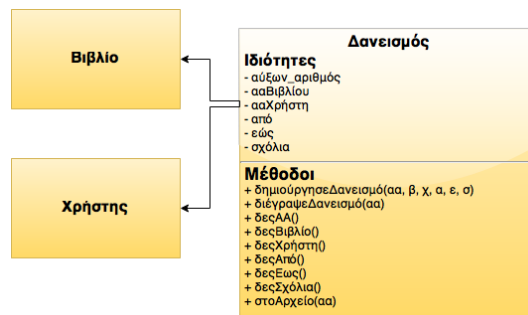
**Σχήμα 4.9:** Συνολική Απεικόνιση του Διαγράμματος Δομής μιας Σχολικής Βιβλιοθήκης

Η παραπάνω σχεδίαση του διαγράμματος δομής του συστήματος είναι καθαρά ενδεικτική. Σε καμία περίπτωση δεν είναι η μοναδική που μπορεί να γίνει, ούτε μπορεί να θεωρηθεί πλήρης από τον κάθε αναγνώστη. Το διάγραμμα αυτό μπορεί να τροποποιηθεί, να εμπλουτιστεί και να μεγαλώσει τόσο κατά πλάτος όσο και κατά ύψος. Αυτό εξαρτάται από πως θέλουμε να οργανώσουμε την λειτουργία της βιβλιοθήκης μας.

### 1.3 Σχεδιασμός προσανατολισμένος στα αντικείμενα

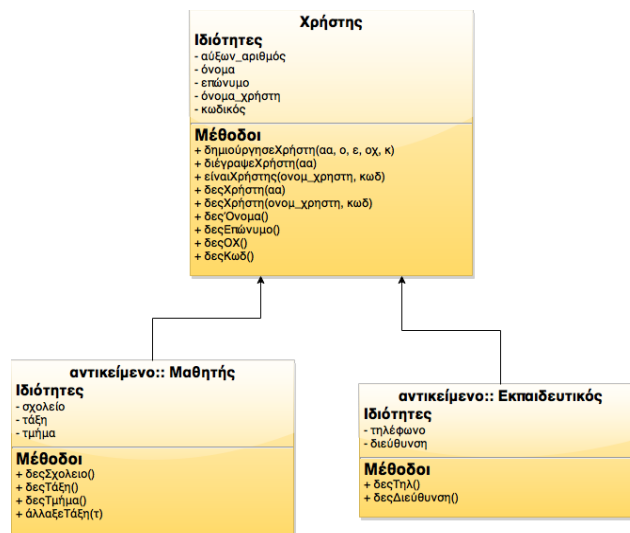
Στο 3ο κεφάλαιο παρουσιάστηκε η αντικειμενοστραφής προσέγγιση στην ανάλυση ενός συστήματος. Συγκεκριμένα προσδιορίστηκε από τι αποτελούνται τα αντικείμενα, τι ακριβώς αντιπροσωπεύουν και πως μπορούν να εντοπιστούν στο πεδίο μιας εφαρμογής. Η φάση του σχεδιασμού ξεκινάει με αφετηρία τα ευρήματα της ανάλυσης που αφορούν τα αντικείμενα που έχουν εντοπιστεί σε ένα σύστημα και τις περιπτώσεις χρήσης. Στη συνέχεια τα αντικείμενα που έχουν εντοπιστεί σκιαγραφούνται με σκοπό να προσδιοριστεί το πως ακριβώς θα ικανοποιήσουν τις απαιτήσεις που υπάρχουν από το σύστημα και μοντελοποιούνται οι κλάσεις των

αντικειμένων. Λεπτομερώς ανατίθενται οι λειτουργίες, που εξάγονται από τις περιπτώσεις χρήσης, στα αντικείμενα και έτσι συνάγονται οι μέθοδοι των κλάσεων. Παράλληλα ορίζονται οι πληροφορίες που είναι απαραίτητο να διαχειρίζονται τα αντικείμενα και προσδιορίζονται οι ιδιότητές τους. Δημιουργούνται τα διαγράμματα των κλάσεων και ταξινομούνται. Αποτυπώνονται οι σχέσεις μεταξύ τους και εφαρμόζονται οι κανόνες της αντικειμενοστραφούς σχεδίασης όπως η κληρονομικότητα.



Σχήμα 4.10 : Η κλάση του δανεισμού

Στο σχήμα 4.10 αποτυπώνεται η κλάση του «Δανεισμού» όπου δείχνει ότι ένα αντικείμενο δανεισμού έχει δύο σαφείς αναφορές. Η πρώτη είναι στο αντικείμενο που αντιπροσωπεύει το βιβλίο που θέλει να δανειστεί ο χρήστης και η δεύτερη στο αντικείμενο που αντιπροσωπεύει τον ίδιο τον χρήστη.



Σχήμα 4.11: Η κλάση του χρήστη

Η κληρονομικότητα χρησιμοποιείται στην αντικειμενοστραφή προσέγγιση όταν υπάρχει μια ομάδα από κλάσεις που έχουν κοινές ιδιότητες και μεθόδους. Στο παράδειγμα της σχολικής βιβλιοθήκης μια τέτοια ομάδα είναι ο χρήστης-μαθητής, ο χρήστης-εκπαιδευτικός κ.α. Σε αυτή την περίπτωση ορίζεται μια γενική κλάση, που ονομάζεται υπερκλάση και από την οποία κληρονομούν τις κοινές μεθόδους και ιδιότητες όλες οι κλάσεις της ομάδας χωρίς να χρειάζεται να τις ορίσουν ή να τις υλοποιήσουν εκ νέου. Στο σχήμα 4.11 φαίνονται η υπερκλάση «Χρήστης» και οι

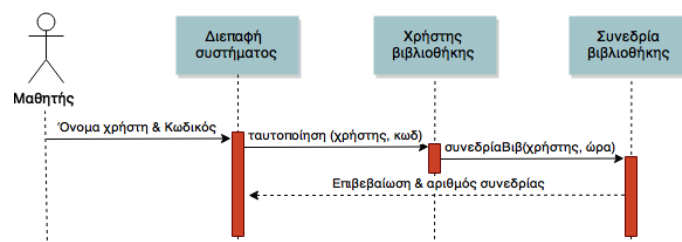
υποκλάσεις «*Μαθητής*» και «*Εκπαιδευτικός*» που κληρονομούν από την πρώτη κλάση. Η κλάση «*Χρήστης*» ορίζει τις κοινές ιδιότητες και μεθόδους που έχουν ο «*Μαθητής*» και ο «*Εκπαιδευτικός*» κι έτσι ο σχεδιαστής δεν χρειάζεται να τις ορίσει εκ νέου στις δυο υποκλάσεις.

Τελικό προϊόν της φάσης αυτής του σχεδιασμού είναι οι μέθοδοι και λειτουργίες των αντικειμένων, το πεδίο ευθύνης τους και ο τρόπος που συσχετίζονται και οργανώνονται.

**Δραστηριότητα:** Σε ένα ηλεκτρονικό σύστημα βιβλιοθήκης κρίνεται σκόπιμο να υπάρχει η δυνατότητα της κράτησης ενός βιβλίου. Να σχεδιάσετε την κλάση της κράτησης. Με ποιες κλάσεις συνδέεται;

### 1.3.1 Διαγράμματα ακολουθίας

Τα διάγραμμα ακολουθίας χρησιμοποιούνται κατά τη διάρκεια της φάσης σχεδιασμού που υιοθετεί το αντικειμενοστραφή μοντέλο προσέγγισης. Απεικονίζουν τον τρόπο με τον οποίο τα αντικείμενα του συστήματος αλληλεπιδρούν μεταξύ τους. Γενικά απεικονίζουν τον κύκλο ζωής των αντικειμένων και τον τρόπο που διεξάγονται τα σενάρια λειτουργικότητας. Δίνουν έμφαση στην χρονική στιγμή που δημιουργούνται αντικείμενα και στην ακολουθία που ανταλλάσσουν μηνύματα για να επικοινωνήσουν μεταξύ τους. Έτσι γίνεται καλύτερη κατανόηση της λειτουργίας των αντικειμένων και επανεξετάζονται τυχόν θέματα που προκύπτουν.



**Σχήμα 4.12:** Διάγραμμα ακολουθίας είσοδος μαθητή στο σύστημα

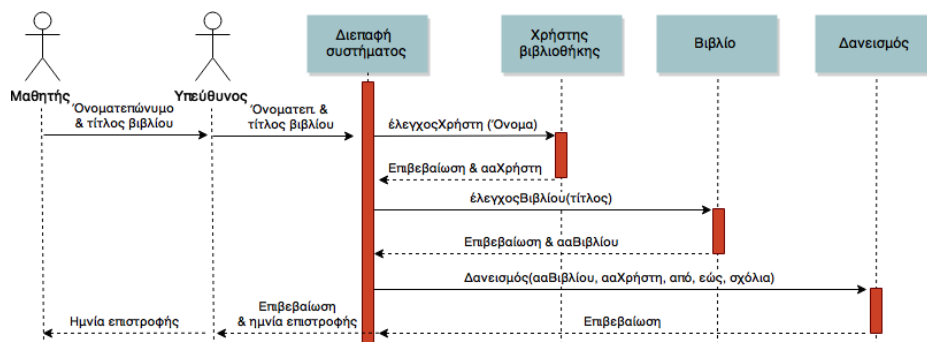
Το διάγραμμα ακολουθίας του σχήματος 4.12 δείχνει την λειτουργία εισαγωγής του μαθητή στο σύστημα της βιβλιοθήκης. Οι κάθετες διακεκομμένες γραμμές αναπαριστούν τον χρόνο ενώ οριζόντια διατάσσονται τα διάφορα αντικείμενα και οι πράκτορες-ηθοποιοί του συστήματος. Στο παράδειγμα φαίνονται ο μαθητής και τα αντικείμενα «*Διεπαφή συστήματος*», «*Χρήστης βιβλιοθήκης*», «*Συνεδρία βιβλιοθήκης*». Τα μικρά ορθογώνια κουτάκια πάνω στις γραμμές χρόνου αναπαριστούν την διάρκεια εκτέλεσης των λειτουργιών από τα αντικείμενα. Τα αντικείμενα επικοινωνούν μεταξύ τους με ανταλλαγή μηνυμάτων που αναπαριστούνται με τα βελάκια.

Ο μαθητής αλληλεπιδρά με το αντικείμενο που διαχειρίζεται την διεπαφή χρήστη και εισάγει τα στοιχεία όνομα χρήστη και κωδικό. Το σύστημα δημιουργεί το αντικείμενο

του χρήστη και ελέγχει την ορθότητα των στοιχείων. Αν τα στοιχεία είναι σωστά τότε δημιουργείται ένα αντικείμενο συνεδρίας με το όνομα του χρήστη και την ώρα εισαγωγής στο σύστημα. Επιστρέφεται στη διεπαφή ο κωδικός-αναγνωριστικό της συνεδρίας και ένα μήνυμα επιβεβαίωσης που πιστοποιεί ότι έγινε κανονικά η είσοδος στο σύστημα. Το αντικείμενο συνεδρίας είναι ενεργό για όσο χρόνο ο χρήστης πλοηγείται στην εφαρμογή. Διαγράφεται με την έξοδο του χρήστη ή μετά από μία ώρα αδράνειας της συνεδρίας.

Στο σχήμα 4.13 φαίνεται το διάγραμμα ακολουθίας δανεισμού βιβλίου στη σχολική βιβλιοθήκη.

**Δραστηριότητα:·** Ποιους ηθοποιούς αναγνωρίζετε στο σύστημα και ποια αντικείμενα. Να εξηγήσετε το σενάριο που απεικονίζεται στο σχήμα 4.12



Σχήμα 4.13: Διάγραμμα ακολουθίας δανεισμού

**Δραστηριότητα:·** Να δημιουργήσετε ένα διάγραμμα ακολουθίας για την επιστροφή ενός βιβλίου στη σχολική βιβλιοθήκη.

## 1.4 Σχεδιασμός διεπαφής χρήστη

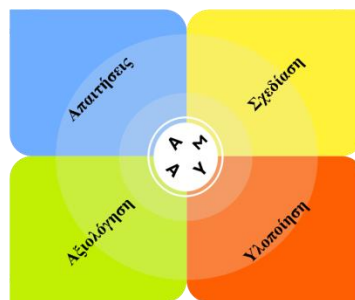
Η διεπαφή χρήστη είναι το τμήμα του πληροφοριακού συστήματος που έρχεται σε επαφή και αλληλεπιδρά με τον χρήστη. Η σωστή σχεδίαση της διεπαφής χρήστη είναι πολύ σημαντική για το πληροφοριακό σύστημα καθώς θα πρέπει να αναδεικνύει όλες τις δυνατότητες που προσφέρει το σύστημα και παράλληλα να ανταποκρίνεται στις ικανότητες, στις ιδιαιτερότητες και στις εμπειρίες του χρήστη.

### 1.4.1 Πρωτότυπα

Μια μέθοδος για την ανάπτυξη της διεπαφής χρήστη είναι η δημιουργία πρωτοτύπων. Τα πρωτότυπα είναι απλοποιημένα μοντέλα του συστήματος που δημιουργούνται με σκοπό την έγκαιρη εκτίμηση κάποιων χαρακτηριστικών που θα υπάρχουν στο τελικό προϊόν.

Τα πρωτότυπα χρησιμοποιούνται σε πολλούς τομείς. Στην αυτοκινητοβιομηχανία είναι μινιατούρες των αυτοκινήτων, στον κατασκευαστικό τομέα είναι οι μικρογραφίες κτιρίων, πάρκων κ.α, στη ζωγραφική είναι τα δοκιμαστικά σκίτσα. Στα πληροφοριακά συστήματα είναι μια αναπαράσταση ενός μέρους του συστήματος.

Για τη δημιουργία των πρωτοτύπων ακολουθείται ένας μικρός κύκλος ζωής εντός της φάσης του σχεδιασμού. Αρχικά εξετάζονται οι απαιτήσεις που έχουν οριστεί στην προηγούμενη φάση τους κύκλου ζωής του συστήματος. Στη συνέχεια ακολουθεί η σχεδίαση του πρωτοτύπου όπου επιλέγονται τα εργαλεία που θα χρησιμοποιηθούν, το είδος πρωτοτύπου που θα αναπτυχθεί, το στυλ και τα χαρακτηριστικά του. Την σχεδίαση ακολουθεί η υλοποίηση όσων προσδιορίστηκαν στον σχεδιασμό με τη χρήση των κατάλληλων εργαλείων. Στο τελικό στάδιο γίνεται η αξιολόγηση του πρωτοτύπου όπου έχει σχέση με τον έλεγχο των χαρακτηριστικών τα οποία προάγει. Η αξιολόγηση ενδεχομένως να αναδεικνύει προβλήματα που θα οδηγήσουν στον πρώτο στάδιο του κύκλου ζωής του πρωτοτύπου και στην εξέλιξή του από την αρχή.



**Σχήμα 4.14:** Κύκλος ζωής πρωτοτύπου

Η χρήση πρωτοτύπων έχει πολλά πλεονεκτήματα. Οι σχεδιαστές έχουν τη δυνατότητα να αναπαραστήσουν τις ιδέες τους με ένα τρόπο που να ευνοείται η επεξεργασία τους και η δημιουργία εναλλακτικών λύσεων. Έτσι διαμορφώνονται οι απαραίτητες συνθήκες για τον έλεγχο των εναλλακτικών λύσεων πριν τη δέσμευση σε κάποια από αυτές και την εξέλιξη του πρωτοτύπου. Εξάλλου η χρήση των πρωτοτύπων δημιουργεί ένα πλαίσιο επικοινωνίας ανάμεσα στους σχεδιαστές και στους ενδιαφερομένους. Οι ενδιαφερόμενοι σε αυτή την περίπτωση μπορεί να είναι οι πελάτες, οι χρήστες του συστήματος και οι υπεύθυνοι αξιολογητές του έργου. Παρέχεται λοιπόν στους ενδιαφερόμενους η δυνατότητα να έχουν έγκαιρα μία αντίληψη σχετικά με τα χαρακτηριστικά και τη φύση του τελικού προϊόντος. Επιπρόσθετα μπορούν να συμμετέχουν στην αξιολόγηση των νέων ιδεών σχεδίασης ή ακόμα και να υποδείξουν λύσεις για την βελτίωσή τους. Ανάλογα με το κόστος και το είδος των πρωτοτύπων και τις συνθήκες συνεργασίας θα μπορούσαν να επιλέξουν από μια σειρά εναλλακτικών πρωτοτύπων και σχεδίων που θα εκφράζουν διαφορετικές προσεγγίσεις και στυλ.

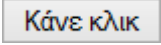
Απόρροια όλων των παραπάνω είναι η ελαχιστοποίηση των προβλημάτων που μπορεί να εμφανιστούν μετά την υλοποίηση του προϊόντος. Τυχόν προβλήματα και θέματα που μπορεί να υπάρχουν είναι πολύ πιθανό να εντοπιστούν κατά τη διάρκεια χρήσης και αξιολόγησης των πρωτοτύπων γεγονός που δίνει ευελιξία στην ομάδα σχεδιασμού να επεξεργαστεί και να αναμορφώσει τις λύσεις και να ελαχιστοποιήσει το κόστος.

#### 1.4.2 Αντικείμενα διεπαφών



Η διεπαφή του χρήστη είναι μια οπτική σύνθεση από γραφικά αντικείμενα τα οποία εμφανίζονται στο οπτικό μέσο και χρησιμοποιούνται για να ικανοποιήσουν την αμφίδρομη σχέση χρήστη και μηχανής. Τα αντικείμενα αυτά χρησιμοποιούνται για τη δημιουργία συμβάντων και για την έξοδο ή είσοδο πληροφοριών.

### Κουμπιά εντολών – ActionButtons:

 Τα κουμπιά εντολών παρουσιάζουν τις βασικές ενέργειες μιας διεπαφής και δίνουν τη δυνατότητα στους χρήστες να τις εκτελέσουν.

```
<button type="button" onclick="alert('Γειασας!')">Κάνε κλικ</button>
```

### Πλαίσια Κειμένου – TextBoxes:

Όνομα:  Τα πλαίσια κειμένου παρέχουν τον βασικό τρόπο για να εισάγουν δεδομένα οι χρήστες πληκτρολογώντας μέσα στο πλαίσιο. Χρησιμοποιούνται και για προβολή δεδομένων δίνοντας παράλληλα τη δυνατότητα επεξεργασίας στον χρήστη.

```
Όνομα: <input type="text" name="firstname">
```

### Κουμπιά Επιλογών – CheckBoxes:

☒ Δίπλωμα αυτοκινήτου  
☐ Δίπλωμα μοτοσυκλέτας  
☒ Δίπλωμα φορτηγού

Τα κουμπιά επιλογών χρησιμοποιούνται για την εισαγωγή δεδομένων τύπου διακόπτη για την εισαγωγή χαρακτηριστικών ή για την μη εισαγωγή αυτών.

```
<input type="checkbox" name="cdl" value="cdl">Δίπλωμα αυτοκινήτου<br>
<input type="checkbox" name="mdl" value="mdl">Δίπλωμα μοτοσυκλέτας<br>
<input type="checkbox" name="mdv" value="mdv">Δίπλωμα φορτηγού
```

### Ράδιο-Κουμπιά Επιλογών – RadioButtons:

☒ Άνδρας  
☐ Γυναίκα

Τα ράδιο-κουμπιά επιλογών χρησιμοποιούνται με τον ίδιο τρόπο που χρησιμοποιούνται τα κουμπιά επιλογών με τη διαφορά ότι θεωρούνται μια ομάδα και κάθε φορά επιλέγεται μόνο μία επιλογή από την ομάδα.

```
<input type="radio" name="s" value="male" checked>Άνδρας<br>
<input type="radio" name="s" value="female">Γυναίκα
```

### Λίστα Επιλογών – ListBoxes:



Η λίστα επιλογών δίνει τη δυνατότητα στους χρήστες να επιλέγουν δεδομένα από μία λίστα επιλογών. Προαιρετικά ο προγραμματιστής έχει την δυνατότητα να ομαδοποιεί τις επιλογές που προσφέρει στον χρήστη.

```
<select>
<optgrouplabel="Γαλλικά αυτοκίνητα">
<option value="p">Peugeot</option>
<option value="r">Renault</option>
</optgroup>
<optgrouplabel="Ιταλικά αυτοκίνητα">
<option value="f">Fiat</option>
<option value="l">Lancia</option>
</optgroup>
</select>
```

### Περιοχή Κειμένου – TextArea:

Η περιοχή κειμένου χρησιμοποιείται, όπως και τα πλαίσια κειμένου, για την είσοδο και έξοδο δεδομένων με τη διαφορά ότι επιτρέπει πολλαπλές γραμμές κειμένου να εισαχθούν ή να παρουσιαστούν.

```
<textarea name="message" rows="10" cols="30">
```

```
Θα ήθελα να δηλώσω τα παράπονά μου για το προϊόν σας...</textarea>
```

### Πλαίσια – Fieldsets:

Τα πλαίσια δεν προσφέρουν κάποια λειτουργικότητα. Παρόλα αυτά είναι σημαντικά γιατί επιτρέπουν την ομαδοποίηση διαφόρων αντικειμένων των διεπαφών και έτσι εξυπηρετούν στον καλύτερο σχεδιασμό μιας διεπαφής. Κάθε πλαίσιο έχει ένα όνομα που το

προσδιορίζει και ονομάζεται «Legend».

```
<fieldset>
<legend>Στοιχεία επικοινωνίας:</legend><br>
τηλ: <input type="text"><p>
fax: <input type="text"><br>
</fieldset>
```

### Μπάρες ολίσθησης – Slider:

Οι μπάρες ολίσθησης επιτρέπουν την εισαγωγή αριθμητικών δεδομένων από ένα μεγάλο πεδίο τιμών.

Βαθμός: `<input type="range" id="a" name="a" value="50">`

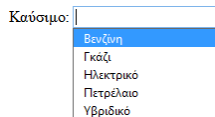
Σημείωση: Για να εμφανίζεται η τιμή δίπλα στην μπάρα θα πρέπει να χρησιμοποιηθεί η εντολή `<output>` και η ιδιότητα `oninput` της φόρμας `<form>`:

`<form oninput="x.value=parseInt(a.value)">`

Βαθμός: `<input type="range" id="a" name="a" value="50">`

`<output name="x" for="a"></output></form>`

### Λίστες δεδομένων – **DataList**:



Οι λίστες δεδομένων επιτρέπουν την εισαγωγή δεδομένων. Καθώς ο χρήστης εισάγει δεδομένα εμφανίζεται μια λίστα επιλογών από την οποία ο χρήστης μπορεί να κάνει μια επιλογή προαιρετικά.

Καύσιμο: `<input list="ka" name="ka">`

`<datalist id="ka">`

`<option value="Βενζίνη">`

`<option value="Γκάζι">`

`<option value="Ηλεκτρικό">`

`<option value="Πετρέλαιο">`

`<option value="Υβριδικό">`

`</datalist>`

### 1.4.3 Βασικές αρχές και λάθη

Ο αισθητικός σχεδιασμός είναι βασικός πυλώνας μιας επιτυχημένης διεπαφής χρήστη. Η καλή αισθητική μιας διεπαφής κεντρίζει το ενδιαφέρον του χρήστη, του προκαλεί όμορφα συναισθήματα και τον προδιαθέτει θετικά. Σχετίζεται με ένα σύνολο παραμέτρων που επηρεάζουν το οπτικό αποτέλεσμα της διεπαφής. Οι πιο σημαντικές από αυτές τις παραμέτρους είναι τα χρώματα, η διάταξη των αντικειμένων και η κατάλληλη μορφοποίηση του κειμένου.

**Δραστηριότητα:** Επισκεφθείτε τις παρακάτω ιστοσελίδες:

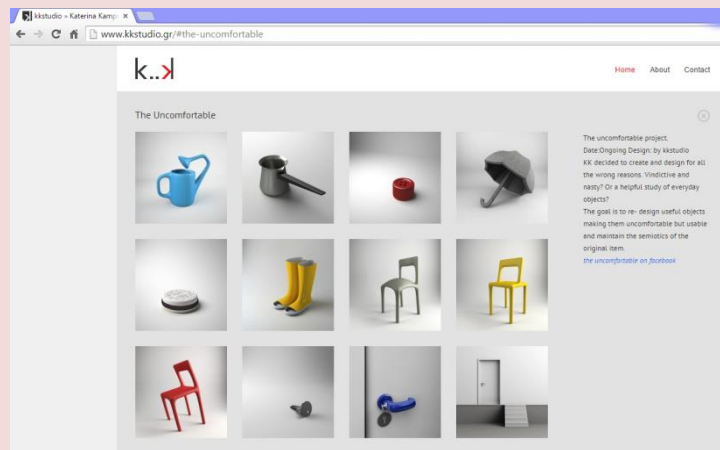
- <http://www.nytimes.com/>
- <http://www.google.gr/>
- <http://www.oracle.com/>

Πως κρίνετε την χρήση χρωμάτων, την διάταξη των αντικειμένων και την μορφοποίηση του κειμένου;

Βρείτε και προτείνετε ιστοσελίδες όπου σας ικανοποιεί ο αισθητικός σχεδιασμός. Εξηγήστε τον λόγο που τις προτιμήσατε.

Ο αισθητικός σχεδιασμός είναι καθοριστικός για την πρώτη εντύπωση που δίνει μια εφαρμογή στον χρήστη. Ωστόσο μια διεπαφή θα πρέπει να είναι χρηστική για να θεωρηθεί πετυχημένη. Αυτό αποτυπώνεται πλήρως σε διάφορα παραδείγματα που βλέπουμε στην καθημερινή ζωή.

**Δραστηριότητα:** Μεταβείτε στην ιστοσελίδα «<http://www.kkstudio.gr/>» της αρχιτέκτων Κατερίνας Καμπράνη και επισκεφθείτε την συλλογή της «The Uncomfortable». Ποια αντικείμενα σας αρέσουν αισθητικά; Βρίσκετε τα αντικείμενα χρηστικά; Ποιο μήνυμα πιστεύετε ότι θέλει να περάσει η σχεδιάστρια με αυτή την συλλογή της;



**Εικόνα 4.15:** Στιγμιότυπο οθόνης από την ιστοσελίδα της Κατερίνας Καμπράνη (τελευταία επίσκεψη 10/7/2015)

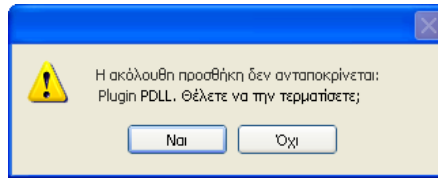
Οι σχεδιαστές διεπαφών πολλές φορές δεν ακολουθούν συγκεκριμένες αρχές αλλά ενεργούν βιωματικά σύμφωνα με τις εμπειρίες τους. Ωστόσο πολλοί ερευνητές, όπως για παράδειγμα οι Jacob Nielsen, Bruce Tognazzini και Ben Shneiderman, έχουν διακρίνει στον σχεδιασμό διεπαφών κάποιες βασικές ιδιότητες και έχουν προτείνει κάποιους κανόνες για την αξιολόγησή τους. Η κατανόηση αυτών των κανόνων και η τήρησή τους κατά τη διάρκεια του σχεδιασμού είναι πολύ πιθανό να οδηγήσουν σε μια επιτυχημένη σχεδίαση.

### Οι κανόνες του Nielsen

Ο Jacob Nielsen διατύπωσε δεκατέσσερις ευριστικούς κανόνες για την αξιολόγηση μιας διεπαφής. Πολλοί από αυτούς τους κανόνες έχουν εφαρμογή σε όλα τα είδη των διεπαφών.

- **Αντιστοιχία μεταξύ του συστήματος και του πραγματικού κόσμου**

Το σύστημα θα πρέπει να ταιριάζει στο νοητικό επίπεδο και την ιδιοσυγκρασία του χρήστη και να ανταποκρίνεται στις προσδοκίες του. Δεν θα πρέπει να χρησιμοποιεί ορολογία και ιδέες που δεν γνωρίζει ο χρήστης:



**Εικόνα 4.16.:** Ένα μήνυμα που δεν καταλαβαίνει ο μέσος χρήστης

Αντίθετα όταν η εφαρμογή ανήκει σε κάποιο συγκεκριμένο πεδίο π.χ. ιατρική δεν θα πρέπει να παραβλέπει την χρήση όρων από αυτή την επιστήμη.

- **Ορατή κατάσταση του συστήματος**

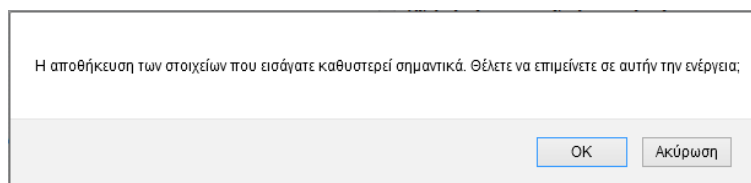
Το σύστημα θα πρέπει να ενημερώνει ανά πάσα στιγμή τους χρήστες για το τι συμβαίνει, με την κατάλληλη ανατροφοδότηση μέσα σε συγκεκριμένο χρονικό διάστημα. Αυτό επιτυγχάνεται με διάφορες τεχνικές όπως η αλλαγή του κέρσορα πάνω σε συγκεκριμένες επιλογές, η ενημέρωση μέσα από τη γραμμή ελέγχου και η επισήμανση αντικειμένων που επιλέγονται.



**Εικόνα 4.17.:** Παραδείγματα όπου η διεπαφή δίνει ανατροφοδότηση για την κατάσταση του συστήματος.

- **Έλεγχος και ελευθερία του χρήστη**

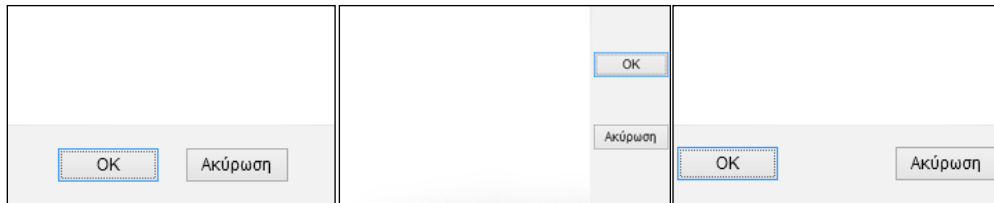
Συχνά δημιουργούνται προβληματικές καταστάσεις στο σύστημα λόγω των χρηστών που κάνουν κάποια επιλογή εκ παραδρομής ή λόγω άλλων αστάθμητων παραγόντων π.χ. αδυναμία σύνδεσης στο Διαδίκτυο. Σε τέτοιες καταστάσεις είναι απαραίτητοι για τους χρήστες τρόποι για να αφήσουν εύκολα και γρήγορα την ανεπιθύμητη κατάσταση.



**Εικόνα 4.18 :** Διάλογος που δίνει στον χρήστη την ελευθερία να παραιτηθεί από μια χρονοβόρα ενέργεια.

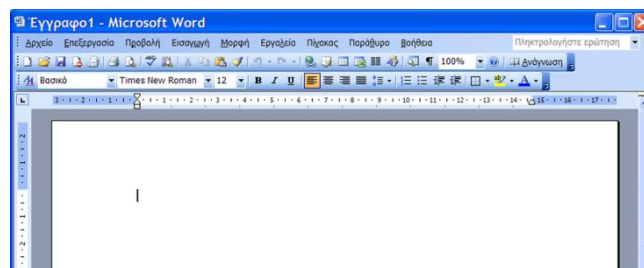
- **Συνέπεια και συμμόρφωση με πρότυπα**

Οι χρήστες δεν θα πρέπει να αναρωτιούνται αν διαφορετικές επιλογές, πληροφορίες ή καταστάσεις σημαίνουν το ίδιο πράγμα.

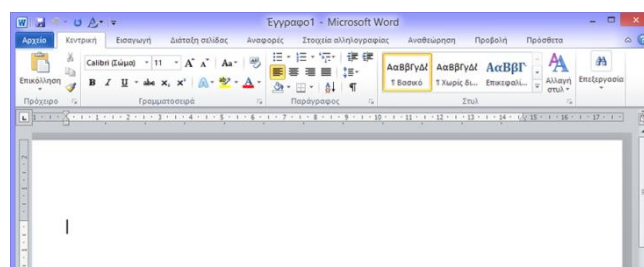


**Εικόνα 4.19.** :Διαφορετικές οθόνες που δείχνουν έλλειψη συνέπειας

Στην εικόνα 4.19 φαίνονται τρεις διαφορετικές οθόνες όπου τα κουμπιά «OK»και «Ακύρωση» τοποθετούνται σε διαφορετικά μέρη της διεπαφής. Η τοποθέτηση των κουμπιών σε διαφορετικές θέσεις και μάλιστα στο ίδιο πρόγραμμα είναι δείγμα έλλειψης συνέπειας και μπερδεύει τους χρήστες.



**Εικόνα 4.20:** Στιγμιότυπο οθόνης του MicrosoftWord 2003

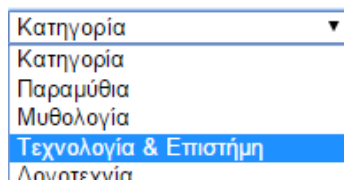


**Εικόνα 4.21:** Στιγμιότυπο οθόνης του MicrosoftWord 2007

Στις εικόνες 4.20 και 4.21 φαίνονται οι οθόνες του MicrosoftWord 2003 και 2007. Η Microsoft παραβίασε τον κανόνα της συνέπειας αφού προχώρησε σε θεαματικές αλλαγές στην διεπαφή από την μία έκδοση του προγράμματος στην άλλη. Στην έκδοση του 2007 το μενού δεν υπάρχει και οι μπάρες που περιέχουν τα εργαλεία έχουν αλλάξει δραστικά.

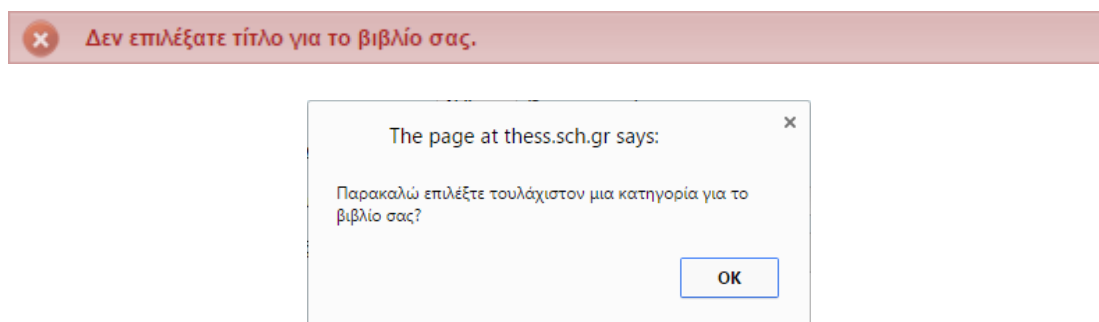
- **Πρόληψη σφαλμάτων**

Το σύστημα θα πρέπει να είναι έτσι σχεδιασμένο ώστε να αποτρέπει τον χρήστη από το να κάνει λάθη. Έτσι για παράδειγμα στην εισαγωγή δεδομένων είναι προτιμότερο για τον χρήστη να επιλέγει από μία λίστα δεδομένων παρά να τα πληκτρολογεί. Τα αντικείμενα των διεπαφών που προσφέρουν έτοιμες επιλογές στον χρήστη όπως η λίστα επιλογών της εικόνας 4.22 είναι προτιμότερα από τα αντίστοιχα αντικείμενα που τον υποχρεώνουν να πληκτρολογεί πληροφορίες όπως τα πλαίσια κειμένου.



**Εικόνα 4.22.:** Το αντικείμενο διεπαφής λίστα επιλογών

Επίσης είναι προτιμότερο ο χρήστης να ελέγχεται όσο το δυνατόν πιο γρήγορα για τυχόν παραλήψεις του. Όπου υπάρχει η δυνατότητα τα σφάλματα θα πρέπει να αναγνωρίζονται εγκαίρως από το σύστημα, να πληροφορούν τον χρήστη (εικόνα 4.23) και να του δίνουν την ευκαιρία να τα διορθώσει πριν προχωρήσει σε κάποια καταχώρηση.



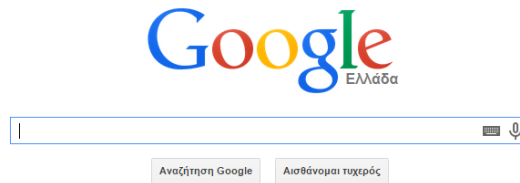
**Εικόνα 4.23:** Μηνύματα που ενημερώνουν τον χρήστη για παραλήψεις

- **Αναγνώριση αντί της ενθύμησης**

Οι επιλογές και όλα τα στοιχεία και θα πρέπει να είναι ορατές ώστε ο χρήστης να μην είναι αναγκασμένος να θυμάται πληροφορίες από ένα τμήμα του διαλόγου σε ένα άλλο.

- **Αισθητική και μινιμαλιστική σχεδίαση:**

Η σχεδίαση θα πρέπει να ακολουθεί τους κανόνες «*Η απλότητα είναι χρυσός*» και «*Οτιδήποτε τοποθετείται στην διεπαφή του χρήστη θα πρέπει να έχει ένα καλό λόγο για να είναι εκεί*». Όλες οι πληροφορίες που περιέχονται στις διεπαφές θα πρέπει να είναι χρήσιμες για τον χρήστη.



**Εικόνα 4.24 :** Η οθόνη του Google

Ένα παράδειγμα εξαιρετικής απλότητας είναι η μηχανή αναζήτησης της Google εικόνα 4.24. Το περιβάλλον του είναι όσο πιο απλό γίνεται. Περιττά αντικείμενα δεν υπάρχουν και χρησιμοποιεί μεγάλη επιφάνεια σε λευκό. Σαν αποτέλεσμα ο χρήστης το θεωρεί πολύ εύκολο στη χρήση, ο σχεδιασμός του φαίνεται κάτι οικείο σαν να το ήξερε πάντα και φορτώνει πολύ γρήγορα.

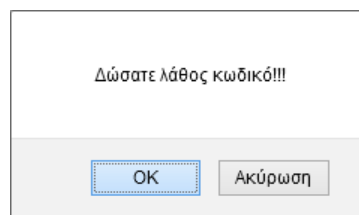


**Εικόνα 4.25:** Δύο διαφορετικές γραμμές εργαλείων

Οι γραμμές εργαλείων δείχνουν τη διαφορά ανάμεσα σε έναν φορτωμένο και ένα μινιμαλιστικό σχεδιασμό. Η πρώτη γραμμή εργαλείων είναι γεμάτη από πολλά φανταχτερά χρώματα που αποσπούν την προσοχή και καθιστούν δύσκολη την ανάγνωσή της. Η δεύτερη γραμμή εργαλείων, από το πρόγραμμα Microsoft Office 2003, χρησιμοποιεί μόνο συγκεκριμένα χρώματα σε κατάλληλους συνδυασμούς. Τα εικονίδια έχουν ομαδοποιηθεί και έχουν κενό ανάμεσά τους έτσι για να διακρίνονται πολύ εύκολα και ακολουθούν τα πρότυπα που είναι γνωστά στον χρήστη.

- **Αναφορά σφαλμάτων**

Το σύστημα θα πρέπει να βοηθάει τους χρήστες να αναγνωρίσουν τα σφάλματά τους, να προβαίνουν σε διάγνωση της αιτίας τους και να ανακάμπτουν από αυτά.



**Εικόνα 4.26:** Μήνυμα σφάλματος

Τα μηνύματα σφαλμάτων θα πρέπει να εκφράζονται σε γλώσσα κατανοητή από τον χρήστη, να εκθέτουν με ακρίβεια το πρόβλημα και όπου είναι δυνατόν να προτείνουν κάποια λύση.

Ευχαριστούμε για την επιλογή σας να μπείτε στην βιβλιοθήκη μας. Είναι φτιαγμένη για εσάς. Έχει φτιαχτεί σε php/mysql για να είναι γρήγορη και αποτελεσματική. Παρόλα αυτά έχουν παρατηρηθεί κάποιες ασυμβατότητες όταν χρησιμοποιείτε έναν όχι και τόσο μοντέρνο web browser όπως για παράδειγμα ο Internet Explorer 6. Υπάρχει ενδεχόμενο η javascript να μην φορτώσει σωστά μέσα στο πρόγραμμά σας. Αυτή δείχνει να είναι και η δικιά σας περίπτωση και γι' αυτό δεν μπορείτε να δείτε τις κατηγορίες του συστήματος. Παρακαλώ κατεβάστε έναν διαφορετικό browser ή απευθυνθείτε στον administrator για περισσότερες πληροφορίες. Να είστε καλά.

OK

**Εικόνα 4.27:** Μια αποτυχημένη αναφορά σφάλματος

**Δραστηριότητα:** Στην εικόνα που ακολουθεί φαίνεται η διεπαφή από το ηλεκτρονικό κατάστημα «Το τέλειο». Η συγκεκριμένη οθόνη παραβιάζει σχεδόν όλου τους κανόνες του Nielsen. Να εντοπίσετε που ακριβώς παραβιάζονται αυτοί οι κανόνες. Να γίνουν προτάσεις για την βελτίωση της χρηστικότητας της διεπαφής.

## Ηλεκτρονικό κατάστημα "Το τέλειο"

Το τέλειο  
Ισμήνης 189, Θεσσαλονίκη  
55535, Ελλάδα  
[Κάνε κλικ για περισσότερες πληροφορίες](#)

### Βήμα 3

Τα νέα μας

[Σας παρουσιάζουμε το νέο Strangeware 47!!!](#)

[Ήρθε για να μείνει: Chilli 254](#)

[Συγκριτικό laptops](#)

[Τα νέα μοντέλα κινητών](#)

[Οδηγός αγοράς tablet](#)

	Item	Qty	Περιγραφή
modify	C1350	1	Κάμερα AH
change	01141	1	Tablet JW
Εισαγωγή			transmit
Σύνολο:			443.99
Μεταφορικά:			16.00
Σύνολο:			459.99

Ευχαριστούμε για την προτίμηση. Σας ενδιαφέρει η εγγραφή στο newsletter μας?

☐ Νέα προϊόντα
 ☐ Προσφορές
 ☐ Δημοπρασίες

☐ Παρακαλώ επιλέξτε
 ☐ Εβδομάδας
 ☐ Μήνα

Διεύθυνση αποστολής

Email:

Όνομα:

Επώνυμο:

Διεύθυνση:

TK:

Περιοχή:

Πόλη:

Τηλ:

Φαξ:

**\*\*Λείπουν στοιχεία από τη φόρμα. Παρακαλώ προσπαθήστε ξανά\*\***

Συνέχισε

**Εικόνα 4.28:** Ηλεκτρονικό κατάστημα «το τέλειο»



Παράδειγμα:

Κανόνας που παραβιάζεται	Σημείο παραβίασης	Πρόταση
Αναγνώριση αντί της ενθύμησης	Στην κεντρική φόρμα της οθόνης με τα αντικείμενα προς αγορά δίνεται η δυνατότητα στον χρήστη να προσθέσει ένα αντικείμενο βάζοντας τον κωδικό στην στήλη «Item» και την ποσότητα στην στήλη «Qty». Σε αυτή την περίπτωση ο χρήστης θα πρέπει να θυμάται τον κωδικό του αντικειμένου που πρέπει να συμπληρώσει στο καλάθι του.	Να διαγραφεί η συγκεκριμένη λειτουργία και να δίνεται η δυνατότητα στον χρήστη να επιστρέφει στη λίστα με τα προϊόντα για να προσθέσει και άλλα αντικείμενα στο καλάθι του.

**Δραστηριότητα:** Αναζητήστε στο Διαδίκτυο τους κανόνες του Bruce Tognazzini και Ben Shneiderman. Βρείτε ομοιότητες και διαφορές με τους κανόνες του Nielsen.

#### **Ερωτήσεις – Δραστηριότητες - Θέματα προς συζήτηση**

1. Δημιουργία διαγραμμάτων δομής για να τεκμηριωθεί ο σχεδιασμός ενός συστήματος.
2. Χρησιμοποίηση του κατάλληλου γραφικού συμβολισμού για την σχεδίαση αντικειμένων συμπεριλαμβανομένων των χαρακτηριστικών και των μεθόδων τους. (Ενδεικτικά παραδείγματα: κατάλογο βιβλιοθήκης, πολυμηχάνημα, σύστημα κρατήσεων).
3. Ομαδοσυνεργατική εργασία με σκοπό την ανάπτυξη μιας πρωτότυπης διεπαφής χρήστη για ένα έργο υιοθετώντας βασικές αρχές σχεδιασμού. Η διεπαφή θα υποστηρίζει τη λειτουργικότητα ενός συστήματος χωρίς όμως να είναι κατά ανάγκη λειτουργικό. Προτείνεται ετεροαξιολόγηση μεταξύ των ομάδων.
4. Να γίνει διάγραμμα δομής μιας υποθετικής σχολικής εικονικής εταιρίας πώλησης αντικειμένων των μαθητών. Να σχεδιασθεί τμήμα παραγγελιών και τμήμα πωλήσεων αντικειμένων.
5. Χρησιμοποιώντας όλα τα αντικείμενα που σας δίνονται στο 4.3 και τους κανόνες που μάθατε στο 4.4 να δημιουργήσετε μια διεπαφή για το ηλεκτρονικό σύστημα εισαγωγής βιβλίων σε μια σχολική βιβλιοθήκη.

### *Βιβλιογραφία*

Στα Ελληνικά

Βεσκούκης, Β. (2000). *Τεχνολογία Λογισμικού*. Πάτρα: Ι. Ε.Α.Π,

Γιακουμάκης Μ. & Διαμαντίδης Ν. (2009). *Τεχνολογία λογισμικού*. Αθήνα: Σταμούλη Α.Ε.

Στα Αγγλικά

Garlan D. & Shaw M. (1994). *An Introduction to Software Architecture*.  
Ανακτήθηκε από  
[http://www.cs.cmu.edu/afs/cs/project/vit/ftp/pdf/intro\\_softarch.pdf](http://www.cs.cmu.edu/afs/cs/project/vit/ftp/pdf/intro_softarch.pdf)

Nielsen J. (1993). *Usability Engineering*. 1st/edition, London: Academic press.

O'Brien J. A. & Marakas G. (2008). *Introduction to Information Systems*.  
14th/edition, New York: McGraw-Hill.

Pfleeger L.S. (2012). *Τεχνολογία λογισμικού Θεωρία και πράξη*. Μτφρ. Φρυσήρας Κ.,  
Επιμ Σταμέλος Γ, Αθήνα: Κλειδάριθμος.

Sommerville I. (2009). *Βασικές αρχές τεχνολογίας λογισμικού*. Μτφρ. Τσιλογιάννης Δ.  
Αθήνα: Κλειδάριθμος.